Delivering heterogeneous services in ICN using Network Slicing

Asit Chakraborti, Ravi Ravindran, Aytac Azgin, Syed Obaid Amin
Huawei Technologies, Santa Clara

IMT 2020/5G Workshop and Demo Day, 2018
Agenda

• ICN Introduction
• Realizing ICN as a Slice
• ICN for Edge Deployment
• Virtual Service Edge Router (V瑟) Platform
• ICN-IoT Requirements
• ICN-IoT Architecture and Prototype
• ICN Conferencing over V瑟
• Conclusion
What is ICN?

- ICN stands for “Information-Centric Networking” [1]
- Continued Networking Evolution
  - Circuits, Packets, Host Connectivity → Information Based Network APIs
- Provides name based abstraction to Application
  - Includes Content, Services and Devices
  - Location Independence of Cache and Compute
- Features: Naming/Security, Mobility, Multicasting, Multihoming, In-Network Computing
- Serves Realtime/Non-Real time, D2D, Ad hoc & IoT Apps.
- CCN/NDN is a popular candidate ICN protocol, though there are others like MobilityFirst, XIA, NetInf etc.
- Currently evolving under IRTF/ICNRG Research Group [2]

ICN Benefits

• Flatter Network architecture - embedded Mobility & Multicasting
• Saving Backhaul Link Cost
• Efficient Edge Compute deployments
  • Always routes to the optimal service point
  • In-network compute function enabled through Named Function Networking (NFN)
• Contextual Trust models [3]
  • Naming and Keys can be contextually related
• Increased reliability on wireless hop
  • Special adaptation layers at the face level (e.g. LTE) to improve reliability
• Allows DataMuling feature in Ad hoc networks [1]
• Memory and Power cost savings in Constrained devices [2]

ICN in a Network Slice [1]

- Realize end-to-end dedicated network for specific service scenario eMBB, URLLC, mMTC.
  - Spans UE, RAT, Transport, Edge Clouds, DCs
- Meet specific service objectives of Security, Latency, QoS, Reliability etc.
- End-to-end virtualization of Compute, Bandwidth, Storage, Data, Device resources.
  - Virtualization allows resources to be efficiently flexibly managed among various slices.
- Specialized Data/Control Plane and Service Control functions to enable rich services.
  - Software Network Functions, P4/POF Platforms
  - Mobility-as-a-service, Security-as-a-service, Context Processing etc.
- Creates scope for new network Architectures like ICN to address 5G Challenges
  - Multi-modal delivery connectivity: M2M, P2P, P2MP and MP2MP
  - Handle Mobility within the Slice
  - New APIs and Service Functions in the Network Architecture

Heterogeneous ICN Slices

- Meet requirements for differing service requirements
- Softwarization of network functions including the ICN forwarder allows better resource isolation with slices
  - Cache can be managed based on application nature
  - FIB is more manageable per slice
  - Different flavors of ICN protocols can be used
- Service and in-network compute functions in ICN get better isolation in slices
- Mobility management per slice
ICN in the Edge

• ICN makes lot of sense in the edge
  – Seamless **contextual networking platform** to connect heterogeneous devices, applications with edge compute, cache and storage resources.
  – Contextual routing to service points
    • E.g. DNS would have scalability problems in a decentralized edge.
  – Efficiently deployed in-network compute with ICN can make proper use of precious edge resource and reduce backhaul bandwidth use.
  – Reusing shareable data via location independent caching both upstream and downstream.
  – Receiver oriented Communication – Multi-homing, Mobility and Multicasting.
  – Challenges: privacy, security, data accountability.
Virtual Service Edge Router (VSER) Architecture [1]

- VSER platform allows to create Service Slices leveraging features such as Name Based Routing, Seamless Mobility Support, Caching, Multicasting and Multihoming.

Multiple ICN Service Slices

- IoT Device-1
- IoT Device-2
- IoT Device-3
- IoT Device-4
- Sensor-1
- Sensor-2
- Actuator
- Camera
- Anchor-Node-1
- Anchor-Node-2
- ICN-Router-1
- ICN-Router-2
- Video Conferencing Slice
- Conference-Client-1
- Conference-Client-2
- IoT Slice
- IoT Client-1
- IoT Client-2
- Network Controller
- IoT Service Controller
- Conference Service Controller
- Orchestrator
- UI
- Docker Registry
- Control plane
ICN for IoT


Considering 50B Things to be connected to the Internet (Heterogeneous, physical things (assets), low power requirements, M2M, Mobile, Ad Hoc etc.)

**Inter-operability**
- Unified Naming of Devices/Services/Content (IPv6 may not be sufficient)
- Flexible Naming (sensors, embedded devices, wearables, smart devices etc.)
- Naming (Persistent (Contextual), Secure, Human friendly)
- Open-API at all levels

**Security, Privacy & Trust**
- Access Control/Trust/Provenance/Data Integrity/Regulations
- Data Privacy/Secure Names

**Scalability**
- ID/Locator Split
- Enable Decentralized Communication (P2P)

**Mobility**
- Devices/Services accessible irrespective of Mobility or Migration

**Reliability/Availability**
- Storage and Caching (Sharing information, reducing upstream bandwidth, Processing)
- Disruption tolerance (QoS, Wireless, Redundancy, Flow Control, Opportunistic transmission)
- Near real-time requirements
- Multi-path & Multi-homing

**Flexibility**
- Heterogeneity (Lossy Radios; Traffic (Push/Pull, Latency, Critical Events))
- Contextual Communication (Varies with IoT application, generally includes Location, Time, Policies etc.)
- Self Organizing (Edges, Simple Networking, Zero Configuration, Minimum Management)
- Adhoc and Infrastructure Mode (Topology/Service Discovery, Routing, Scalable Name Resolution)

**Management**
- FCAPS to IoT Services, Network, Devices, Protocols
- Scale to Large Number of Devices
- Requirements management of in-network Content, Services

Requirements vary with the Scenario such as Health Care/Smart Grid/Transportation/Home Networks/Industrial etc.
ICN IoT Slice Operation

- **Node distinction**
  - IoT network
    - Resource constrained IoT nodes with sensors and actuators
    - Aggregator node with more resources
  - Network slice ISR components
    - Authentication Manager
    - IoT Agent
    - IP gateway/server

- **All components communicate using lightweight ICN protocol (ccn-lite)** [1]

- **Discovery**
  - The new nodes serve /service-discovery interests with data indicating the details about its services and sensor/actuator details
  - discover IOT services in its network by sending /service-discovery interests with exclusion mechanism

- **Policies**
  - IoT Agent maintains user policy settings
  - Handles IoT data and logs, executes policies
  - Can scale up with more users

ICN IoT device onboarding

• Secure onboarding of ICN IoT devices is a challenge, our model is based on LASer a 3 phase protocol [1]

• Network discovery
  — New IOT devices and IoT service share a Key before onboarding (Pre-shared Key)
  — New devices send a /discover interest to verify if it can trust the network and also to find its next hop
    • Include ID, Nonce, distance from Aggregator Node (AN)
  — Renamed by neighbor that serves it to include its own information (MAC) and signature
  — Destined for Authentication Manager (AM)
  — AM derives the new devices’ Authentication Key (using Pre-shared Key and the new device ID present in the message) and uses it to generate signature, also including a nonce in the data
  — The new device verifies the signature and extracts the data
  — The response tells the new device its next-hop for unicast packets

• Device Authentication
  — Then the new devices send information about itself so that AM can verify it
  — Include previous nonces, IDs of all parties, signed using its Authentication Key
  — AM verifies signature, responds with Routing Key (signed and encrypted by the Transient Keys for the new devices that it derived) that should be used inside the AN’s network

• Path advertisement
  — New IOT device use a set-next interest to allow upstream nodes to set routing entries towards it
  — Provide own ID and that of neighbor, and interface information
  — Every node till AM processes it to set downwards FIB (with device ID and next hop MAC) and resends it upwards
  — Device ID is used for routing

Onboarding timeline

- **Discovery**
  - \(I:/\)discover/<IDRNN>/<RNN>/<ADRNN>
  - \(I:/\)discover/<IDRNN>/<RNN>/<ADRNN>
  - **broadcast**
  - \(D:/\)IDRNN || RNN || IDAM || RAM || MACRID || ADRNO || IDAN]RKKNN

- **Authentication**
  - \(I:/\)onboard/<IDRNN>/<RNN>/<MACRID>/ADRNO/IDAN]RAKAN
  - \(I:/\)auth/<IDRNN>/<RNN>/<RAM>/IDAN]RAKAN

- **Routing Advertisement**
  - \(I:/\)set-next/<IDRNN>/<MACRID]RAKAN
  - \(I:/\)set-next/<IDRNN>/<MACRID]RAKAN
  - \(I:/\)set-prefix/<IDRNN>/<IDAN]RAKAN

\(RN_{\text{NEW}}\) \(\rightarrow\) \(RN_{\text{OLD}}\) \(\rightarrow\) \(AN\) \(\rightarrow\) \(AM\)
Serverless Scalable Audio-Video Conferencing over VSER [1-3]

Edge Cloud based ICN A/V Deployment

Edge Cloud based ICN A/V Solution

• Conference Controller Functions
  - Enable MP-2-IP Connectivity
  - Conference Level Virtualization: Multiple Simultaneous Conferences, Service Scaling, Dynamic Name Based Routing, Conference Monitoring and Management.
  - Context level Adaptation

• Current solutions such as Skype, Goto-Meeting, Webex follows a client-server model and are made to scale restricting the number of active producers of media.
• CCN/NDN has to emulate PUSH behavior to meet realtime application requirements.
• Ad hoc participant joint requires immediate synchronization among producers and consumers.
• The bottleneck in our design is the VSER because of unicast towards the participants and consumer due its producer state tracking algorithms.

ICN A/V Conferencing Evaluation

Test Bed for Evaluation

Set Up:
- 3 VSER and Host Nodes (Intel – i7 family)
- Participants emulated in Containers
- End-to-end IP Latency (15)ms

O(N) growth instead of O(N^2)
From 3 → 15 Participants:
~1 → 8 Mbps
(8X Instead of 25X)

→ For 15 All Party Conferencing mostly < 150ms and 250ms for Audio/Video

→ For 40 producers and 1 Consumer Conferencing mostly < 150ms and 250ms for Audio/Video

Conclusion/Future direction

1. Edge computing based on host based networking
2. ICN slice (edge processing at the network layer – caching/aggregation)
3. ICN service slice (mobility/contextual routing to service)
4. In-network compute (NPU/GPU, dynamic compute allocation): including but not limited to NFN [1]/NFaaS [2]

**NFN Characteristics:**
- Goal is to delegate compute to stateless functions in-network
- Client requests contain expression with data and function name, NFN nodes can compute, forward or separate name and data and then forward
- Can pull compute towards data path, or push execution towards function repositories, functions move synchronously to client requests
- **Challenges:** New APIs, Latency

**NFaaS Characteristics:**
- Goal is to allow stateless service functions in-network and move them in strategic directions (i.e., towards edge)
- Routing based on service names, services figure out how to get data based on ICN principles
- Network layer uses context of passing data and heuristics to determine which functions to host
- Functions move asynchronous to client requests, using NDN like pull based mechanism
- **Challenges:** Unikernel acceptance

---

[1] named-function.net
Thank You and Questions ?